

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A) 平3-54649

⑤ Int. Cl.⁵

識別記号

庁内整理番号

⑬ 公開 平成3年(1991)3月8日

G 06 F 12/12

A

7232-5B

審査請求 未請求 請求項の数 1 (全7頁)

⑭ 発明の名称 バッファ記憶制御方式

⑰ 特 願 平1-188744

⑱ 出 願 平1(1989)7月24日

⑲ 発 明 者 阿 部 雅 彦 東京都港区虎ノ門1丁目7番12号 沖電気工業株式会社内
 ⑳ 出 願 人 沖電気工業株式会社 東京都港区虎ノ門1丁目7番12号
 ㉑ 代 理 人 弁理士 鈴木 敏明

明 細 書

1. 発明の名称

バッファ記憶制御方式

2. 特許請求の範囲

目的の情報の写がバッファ記憶装置中に存在しない読み出しミス時には主記憶装置にアクセスして目的の情報をバッファ記憶装置に格納し、

主記憶装置中の情報の写がバッファ記憶装置中に存在するとき行われる写に対する書き込み時にはバッファ記憶装置内の情報のみを更新し、

情報をバッファ記憶装置から追い出す時にその情報を主記憶装置に書き戻すようにしたバッファ記憶制御装置において、

バッファ記憶装置中の情報の最小アクセス単位毎に該当する情報が廃棄可能かどうかを記録する手段と、

廃棄可能かどうかの記録を

イ. 読み出しミス時には真に初期化し

ロ. 書き込み時には偽にし

ハ. 廃棄指示が行われたときには真にし

ニ. その他の場合は状態を保持する

ように制御する手段と、

バッファ記憶装置から情報を追い出す際に、該当する廃棄指示記録が真であったら、主記憶装置への書き戻しを抑止する手段と、

命令中において、バッファ記憶装置の読み出し参照後、その情報について上記廃棄指示を行なう方法を持つ命令セットとを備えていることを特徴とするバッファ記憶制御方式。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、計算機のバッファ記憶装置の制御方式に係り、特にバッファ記憶装置における主記憶装置への書き戻し方式のなかのストアイン方式を改善したものに関する。

〔従来の技術〕

情報処理システムにおいて、第5図に示すように、主記憶装置51に記憶する情報に対する演算装置53からのアクセス速度を実質的に高速化す

る手段として、高速で小容量のバッファ記憶装置（キャッシュメモリともいう）52を設ける方式が採用されている。

この方式においては、アクセスした主記憶装置上の情報の写をバッファ記憶装置に保持する。その後の主記憶アクセス要求において、目的の情報の写がバッファ記憶装置にある場合には、主記憶装置にアクセスすることなく、バッファ記憶装置上の情報を使用する。これにより、実質上バッファ記憶装置のアクセス速度で主記憶装置の情報を得ることができる。

上記のアクセス高速化方式は、バッファ記憶装置内に写のある情報に対して書き込みが行われた場合の制御方式で2つに分けられる。1つは書き込みのたびにバッファ記憶装置と主記憶装置の両方の情報を更新する方法である。もう1つは書き込み時点ではバッファ記憶装置内の情報のみを更新し、かつ更新が行われたことを記録しておくだけで、その情報の追い出しが必要になるまで主記憶装置への書き戻しを遅らせる方法である。前者

域は手続実行中は有効な情報を保持するが、手続から復帰した時点で意味を失う。このことはC言語の他にFORTRANやPASCALなどコンパイラが記憶領域管理するものについて共通にいえることである。

このように従来のストアイン方式のバッファ記憶装置では、スタック中への書き込みが行われると、手続からの復帰後もその情報が残るため、追い出しが必要になった場合に、不要な書き戻しが発生していた。

本発明の目的は、バッファ記憶装置に記憶される情報に主記憶装置への書き戻しの要／不要情報を付することによって、上述した従来技術の欠点を解消して、ストアイン方式のバッファ記憶装置において発生する主記憶装置への不要な書き戻しをなくすようにしたバッファ記憶制御方式を提供することにある。

〔課題を解決するための手段〕

本発明のバッファ記憶制御方式は、目的の情報の写がバッファ記憶装置中に存在しない読み出し

をストアスルー方式、後者をストアイン方式（またはスワップ方式）と呼ぶ。ストアイン方式は、主記憶装置へのアクセス回数を減少させる利点がある。

〔発明が解決しようとする課題〕

上述したようにストアイン方式はバッファ記憶装置上だけで書き込みを完了させてしまう方式であり、その代わり情報を追い出すときには、その内容を主記憶装置上に書き戻す必要がある。

しかし、従来のものでは、バッファ記憶装置から情報を追い出すときに、その内容を一律ないし機械的に主記憶装置に書き戻していたので、実際には主記憶装置に書き戻す必要のない情報まで書き戻すという欠点があった。

例えばミニコンピュータやユニックスで主に用いられるコンパイラ言語の一種であるC言語のようなコンパイラ言語を用いた場合、手続呼び出しのたびに、戻り番地を保存するためのレジスタの退避域や演算レジスタの内容を保存するための一時変数域等がスタック中に確保される。この領

ミス時には主記憶装置にアクセスして目的の情報をバッファ記憶装置に格納し、主記憶装置中の情報の写がバッファ記憶装置中に存在するとき行われる写に対する書き込み時にはバッファ記憶装置内の情報のみを更新し、情報をバッファ記憶装置から追い出す時にその情報を主記憶装置に書き戻すように構成されている。

このような構成のバッファ記憶制御方式において、バッファ記憶装置中の情報の最小アクセス単位毎に該当する情報が廃棄可能かどうかを記録する手段を備えている。

また、廃棄可能かどうかの記録を

イ、読み出しミス時には真に初期化し

ロ、書き込み時には偽にし

ハ、廃棄指示が行われたときには真にし

ニ、その他の場合は状態を保持する

ように制御する手段を備えている。

そしてさらに、バッファ記憶装置から情報を追い出す際に、該当する廃棄指示記録が真であったら、主記憶装置への書き戻しを抑止する手段と、

命令中において、バッファ記憶装置の読み出し参照後、その情報について上記廃棄指示を行なう方法を持つ命令セットとを備えて構成されたものである。

〔作用〕

バッファ記憶装置中の情報がアクセスされると、制御手段により次のように廃棄可能かどうかの判断がなされ、その結果が記録手段に記録される。

まず、バッファ記憶装置からの読み出しミス時にはその情報についての廃棄可能かどうかの判断記録は“真”に初期化され、書き込み時には“偽”にされる。読み出しミス時に“真”にして廃棄可能とするのは、その情報が主記憶装置の情報と同じだからである。また、書き込み時に“偽”とするのは、その情報が主記憶装置の情報と異なるものとなり、書き戻しが必要となるからである。

次に、手続きが呼び出されるような場合においては、スタック中に確保されて手続き実行中に有効となる情報が手続き復帰後は意味を失うことが、予め命令セットによって把握されるため、このよ

うな意味を失う情報についての記録は廃棄指示が行われる。

そして、その他の場合は廃棄可能かどうかの記録が更新されることなく保持される。

従って、バッファ記憶装置から情報の追い出しが必要になった場合に、追い出し情報についての廃棄可能かどうかの記録が抑止手段によってチェックされ、記録が真であれば書き戻しが抑止される。このため、主記憶装置への不要な書き戻しが発生することがない。

〔実施例〕

以下、マッピング方式としてダイレクトマッピング方式を採用したストアイン方式のバッファ記憶装置を例として本発明の実施例を第1図～第4図を用いて説明する。

この例においては、バッファ記憶装置のブロックサイズを16バイト、最小アクセス単位をバイト、バッファ記憶装置と情報処理装置の間での転送単位を4バイト(1ワード)とする。

第2図はバッファ記憶装置52の構成をブロック図で示したものである。バッファ記憶装置52は、目的とする情報がバッファ記憶装置内にある確率を高めるために、主記憶装置から情報を読み出すときに目的の情報だけでなく、その前後の情報も一緒に読み出してこれを記憶する(この単位をブロックと呼び、既述したように本実施例ではこのブロックサイズが16バイトである)。

バッファ記憶装置52は大きく分けて、ディレクトリ部21、制御部22、データ部23の3つの要素で構成される。

データ部23は主記憶装置の情報の写を記憶する。データ部23とディレクトリ部21はそれぞれブロックに分割され、ブロック毎に1対1に対応する。ディレクトリ部21はデータ部23の各ブロック対応に制御情報を保持する。制御部22は与えられたアドレス、ディレクトリ部21の情報をを用いて、バッファ記憶装置52のヒット/ミスの判定(バッファ記憶装置内に目的の情報が見付かるか、見付からないかの判定)、情報の置き

換え、ディレクトリ情報の更新等を制御する。

このように構成されているバッファ記憶装置において、本実施例では、特にディレクトリ部21において保持する情報と、書き戻し制御の方法とに特徴を持たせている。

第1図に本実施例の特徴となる部分の詳細構成を示す。本構成では、ディレクトリ部21はページ番号12、有効(Valid)ビット13(有効のとき“1”無効とき“0”)、廃棄ビット14(d15～d0)から成る。この廃棄ビット14が本発明の廃棄可能かどうかを記録する手段となる。廃棄ビット14は第3図にも示すように、ブロックの1バイト毎に1ビット存在し、1ブロックで16ビットある。これらのビットはブロックを追い出す際に、主記憶装置への書き戻しが必要かどうかを指示する。このビットが“1”のとき、ブロック中の対応するバイトは書き戻す必要がない。

廃棄指示ビット14は以下のような状況のとき第2図の制御部22によって更新される。この制御部22が本発明の記録を制御する手段となる。

①目的の情報にアクセスする場合に、その情報がバッファ記憶装置中になければ主記憶装置にアクセスして目的のデータを含むブロックで読み出してバッファ記憶装置中に格納するとき、即ち主記憶装置からバッファ記憶装置へ新たに写を取るとき、

(a) アクセスが読み出しの場合は全て“1”。バッファ記憶装置に写した情報は新たなものであり、この情報を読み出す分には情報内容に変化はなく主記憶装置の情報と同じであるから、廃棄しても構わないことによる。

(b) アクセスが書き込みの場合は、書き込み位置に対応するビットのみ“0”とする。本来“1”とするべきであるが、(a)で説明したように読みだし時に“1”に更新されるから、むしろ次の②の書き込み時との整合を図るため“0”としている。他は“1”とする。

②目的の情報にアクセスする場合に、その写しが既にバッファ記憶装置中にあり、その写しを更新する時、即ちバッファ記憶装置でヒットした書き

込み時、

書き込み位置に対応するビットを“0”とする。主記憶装置内の情報は古いままだから、最新の情報を主記憶装置へ写す必要があることによる。他は変更しない。

③目的の情報にアクセスする場合に、その写しが既にバッファ記憶装置中にあり、その写しを読み出す時、即ちバッファ記憶装置でヒットした読み出し時で、廃棄指示があった場合、

読み出し位置に対応するビットを“1”とする。他は変更しない。

以上の結果、真に主記憶装置への書き戻しが必要な場合のみ、該当するブロックのバイトの廃棄指示ビットが“0”となっていることになる。従って、ブロックの追い出しを行なう際に、主記憶装置への書き戻しをするかどうかを決定する論理は次のようになる。

WriteBack = (Valid · SamePage*) · (d15
... · d0*)

*印は負論理を示す。

ここで各信号は第1図の符号と次のように対応する。

26: WriteBack : 主記憶装置へ書き戻すことを指示する。“1”のとき書き戻す。

22: SamePage : 与えられたアドレスのページ番号10とディレクトリ部21のページ番号12との比較結果を示す。“0”のとき一致している。

23: Valid : 該当ブロックの情報が有効か無効かを示す。“1”の時有効。

24: d15 ... d0 : バイト毎の廃棄指示ビット。“1”の時書き戻し不要。

以上の論理により、有効な情報の格納されたブロックを追い出す必要があるとき、前記情報が廃棄指示されていたならば主記憶装置への書き戻しが抑止されることになる。

さて、第1図をさらに具体的に説明する。

アドレスの下位ビットに設けたエントリ番号11によりディレクトリ部21のブロックの1つが

選択されるようになっている。ディレクトリ部21のページ番号12はアドレスの上位ビットに設けたページ番号10と比較すべく比較器15に入力される。また16ビット構成の廃棄ビット14はこれらビットのいずれかが“0”のとき非廃棄出力、即ち“1”出力を出すNAND回路16に入力される。このNAND回路16の出力と有効ビット13とは、AND回路17に入力され、そして比較器15の出力は反転した上で同じくAND回路17に入力され、これら入力がすべて“1”のときWriteBack信号“1”を出力するようになっている。

上記比較器15、NAND回路16、AND回路17で、本発明の書き戻しを抑止する手段を構成している。

さて、上記のような構成において、与えられたアドレスの下位ビットであるエントリ番号11によってディレクトリ部21の1ブロックが選ばれる。ディレクトリ部21からは選ばれたブロックのページ番号12、有効ビット13、廃棄ビット

14がディレクトリ部21から出力される。ブロックのページ番号12はアドレスの上位ビットであるページ番号10と比較器15により比較される。もし一致してSamePage信号22が“0”となれば、Valid信号23が“1”である場合に、AND回路17からNAND回路16の出力をWriteBack信号26として出力する。即ち、有効な情報の格納されたブロックをバッファ記憶装置から追い出す必要があるとき、前記情報が廃棄指示されていたならば、換言すると追い出されるブロックの全ての廃棄指示ビット信号24がオール“1”であれば、主記憶装置への書き戻しが抑止される。これに対し廃棄指示ビット信号24の少なくとも一つが“0”であれば、書き戻しが必要であるとして主記憶装置への書き戻しが実行される。

一方、比較器15の比較結果が一致していない場合には、バッファ記憶装置中に目的とする情報が格納されていないため、主記憶装置から目的とする情報を読み出す。この場合、アドレスの上位ビットであるページ番号10と主記憶装置から読

戻しを抑止する(ステップ502)。逆に“0”が一つでも含まれていれば、主記憶装置への書き戻しを行う(ステップ503)。

ここで、コンパイラによる廃棄指示について説明する。コンパイラは、実行時における情報の記憶領域割り付け及びその操作方法を指示する。この記憶領域管理の方法は、C言語にあっては動的割り付け方法がとられる。即ち、プログラム実行時に記憶領域を管理する。このためにスタック技法を使用して、記憶領域はスタック上にとられ、ブロックの入口/出口で割り付け/解放がなされる。

従って、手続き呼び出しのたびにスタック中に確保されるレジスタの退避域、一時変数域等については、その割り付け/解放を予めコンパイラがこれを知ることができ、そのような手続きの場合には、廃棄指示ビットに“1”を立てるよう指示することが可能となる。このように命令中において、記憶装置の読み出し参照後、その情報の廃棄指示を行なう方法を持つ命令セットを用意する。

み出してきた情報を、バッファ記憶装置の対応したブロックに前記廃棄指示ビットも含めて書き込む。

次に、第4図(a)(b)を用いて本実施例の作動をコンパイラによる書き戻し制御を含めて説明する。

第4図(a)においてバッファ記憶装置からの目的の情報の読み出しミス時には、廃棄ビットを真、即ち“1”に初期化し(ステップ401、402、405)、書き込み時には偽、即ち“0”にする(ステップ401、404)。

そして、コンパイラにより廃棄指示が行われたときには廃棄ビットを“1”にし(ステップ403、406)、その他の場合は状態を保持する(ステップ407)。

第4図(b)において、バッファ記憶装置のブロックを追い出す際に、該当する廃棄指示記録が全て“1”か、あるいは“0”が一つでも含まれるかを判断する(ステップ501)。廃棄指示記録が全て“1”であったら、主記憶装置への書き

その結果、手続き呼び出しのたびにスタック中に確保された領域への書き込みが行われ、手続きからの復帰後にその情報が残っていても、追い出しが必要になった場合に、必要なもののみ書き戻すことが可能となる。

なお、本実施例では、主記憶装置とバッファ記憶装置とのブロックの対応付けであるマッピング方式をダイレクトマッピング方式とした場合について説明したが、本発明はフルソシアティブ方式やセットアソシアティブ方式等他の方式にも適用できる。

[発明の効果]

以上説明したように本発明によれば、スタックの情報のように、その後意味を失うような書き戻し不要な情報については、更新が行われても主記憶装置への書き戻しは行なわれなくなる。従って、追い出し時の処理が減り、主記憶装置のバスのトラヒックが減少し、システム性能を向上することができる。

4. 図面の簡単な説明

第1図は本発明によるバッファ記憶制御方式を実施するための構成例を示すバッファ記憶装置のディレクトリ部周りのブロック図、第2図はバッファ記憶装置の基本構成図、第3図はブロックにおける廃棄ビットの説明図、第4図は本実施例の廃棄ビット・書き戻し制御の動作説明図、第5図はバッファ記憶装置の原理図である。

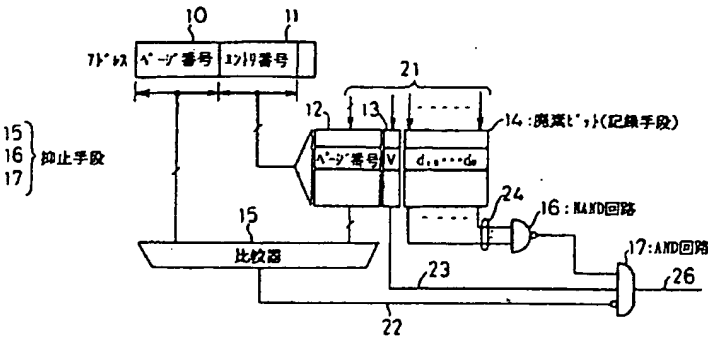
14は情報が廃棄可能かどうかを記録する手段であるディレクトリに設けられた廃棄ビット、15、16、17は主記憶装置への書き戻しを抑止する手段を構成する比較器、NAND回路、AND回路、22は廃棄可能かどうかの記録を制御する手段、23はバッファ記憶装置、51は主記憶装置である。

| ブロック No. | 16バイト(1ブロックサイズ) | | | | | | | | | |
|-------------|-----------------|-----------------|-----------------|-----------------|-----|----------------|----------------|----------------|---|--|
| | d ₁₅ | d ₁₄ | d ₁₃ | d ₁₂ | ... | d ₂ | d ₁ | d ₀ | | |
| 0 | 1 | 1 | 1 | 1 | ... | 0 | 0 | 1 | 1 | |
| 1 | 0 | 1 | 0 | 0 | ... | 1 | 0 | 1 | 0 | |
| 2 | | | | | ... | | | | | |
| 3 | | | | | ... | | | | | |
| ... | | | | | ... | | | | | |
| | | | | | ... | | | | | |

ブロックにおける廃棄ビットの説明図

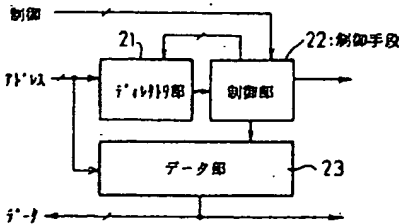
第3図

出願人 沖電気工業株式会社
代理人 弁理士 鈴木 敏 明



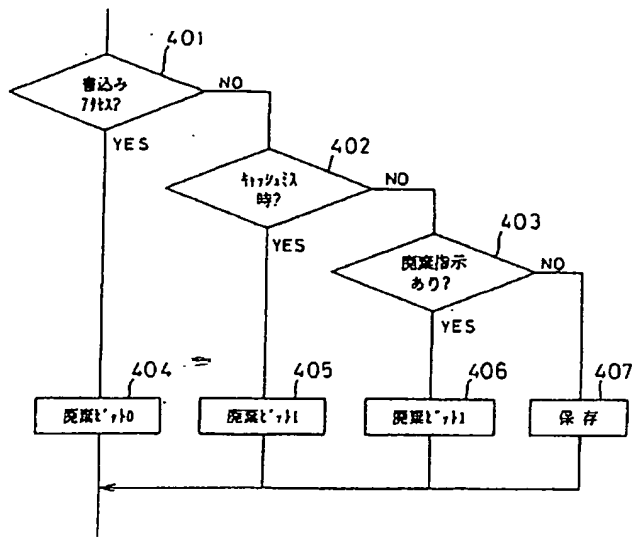
本実施例のnバイト記憶制御方式の主構成

第1図



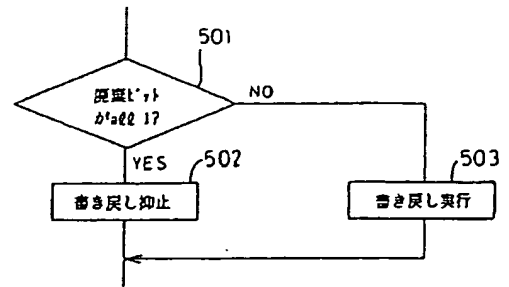
nバイト記憶装置の基本構成

第2図



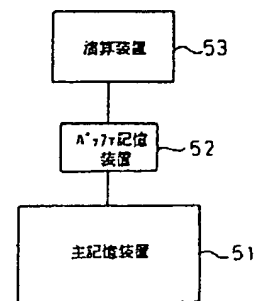
本実施例の履歴リセット制御の動作説明

第4図(a)



本実施例の書き戻し制御の動作説明

第4図(b)



メモリ記憶装置の原理図

第5図

Japanese Laid-Open Patent Application No. 03-54649

Next, the operation of the embodiment including the write-back control by a compiler is described with reference to FIGS. 4 (a) and 4 (b).

In FIG. 4(a), when a miss occurs in the reading of the target information from a buffer memory, the discard bit is initialized to "True", in other words, "1" (step 401, 402 and 405), and when such target information is written, the discard bit is initialized to "False" or, in other words, "0" (step 401 and 404).

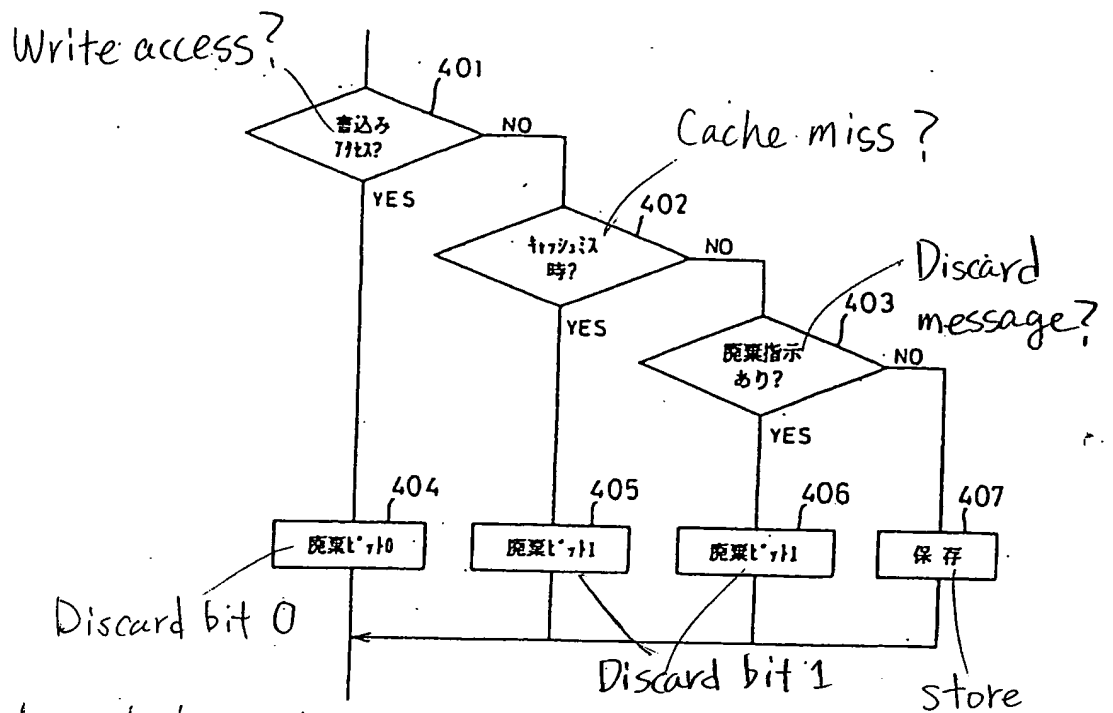
When the discard message is given by a compiler, the discard bit is set to "1" (step 403 and 406). In other cases, the condition is maintained (step 407).

In FIG.4 (b), during an eviction of a buffer memory block, it is judged whether all the corresponding discard message logs are "1" or whether those logs include a "0" (step 501). If all the discard message logs are "1", a write-back to the main memory is not executed (step 502). On the other hand, if a "0" is included, a write-back to the main memory is executed (step 503).

Here, a discard message by a compiler is described. A compiler instructs the allocation of information to a memory area and its operation method at the time of execution. This memory area management method provides dynamic allocation in the C language. More specifically, a memory area is managed when a program is executed. To this end, a memory area is obtained in the stack by using the stack technique and is allocated/released at the entry/exit of a block.

Therefore, the compiler can know the allocation/release of the temporary variable area and the register save area secured in the stack every time the procedure call is executed. In such procedure, it is possible to give an instruction to set "1" to the discard message bit. In this manner, after reference to a read-out from the main memory during the execution of such instruction, the command set that has the method for instructing the discarding of such information is prepared.

As a result, a write operation to the area reserved in the stack is executed every time a procedure call occurs. Even if such written information remains after returning from the procedure, it is possible to write back only necessary information when eviction is required.

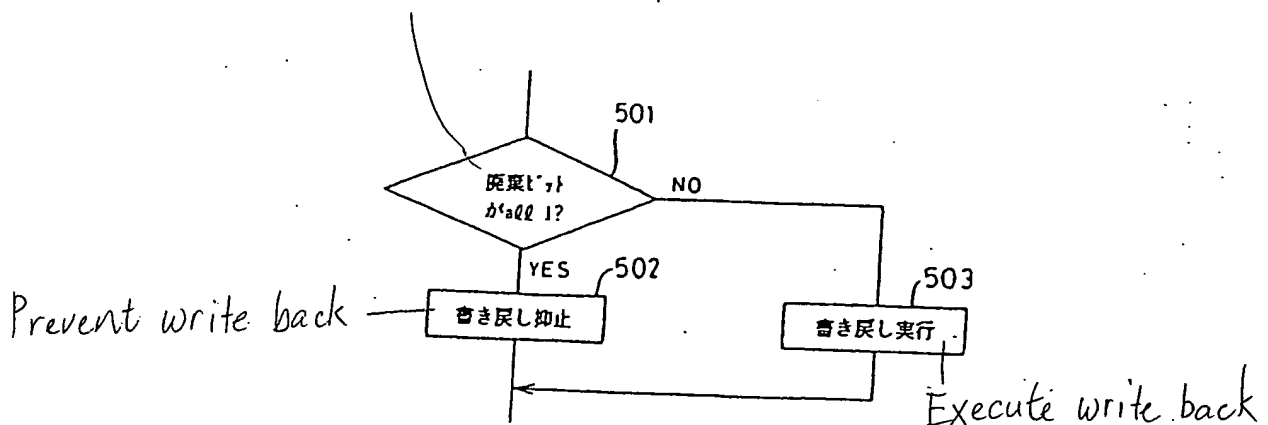


Description of write-back control operation in the present embodiment (本実施例の廃棄ビット制御の動作説明)

第4図(a)

FIG. 4 (a)

All discard bits are "1" ?



本実施例の書き戻し制御の動作説明

第4図(b)

Description of write back control operation in the present embodiment

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.